EXHIBIT D11

Best Practice

Quality Assurance of Product Development in the Lottery Industry: Development Process

April 2004

NASPL

THE Open GROUP

Best Practice

Quality Assurance of Product Development in the Lottery Industry:
Development Process

Document Number: BP0402

Published by The Open Group, April 2004.

Comments relating to the material contained in this document may be submitted to:

bestpractices@opengroup.org

Contents

Preface

North American Association of State and Provincial Lotteries (NASPL)

NASPL has approved the creation of a standards initiative, which is dedicated to the adoption or creation of Technical Standards, Best Practices, and Certification Programs that will further the lottery objectives of integrity, security, interoperability, and profitability.

The NASPL Standards Initiative (NSI) was approved and funded by NASPL and the vendor community as a collaborative development effort with participation from the lotteries, gaming vendor, and retail associations. Project management and facilitation services for standards development and certification are provided by The Open Group.

The NSI Vision is to provide an interoperable lottery environment that is based on a set of open Technical Standards, approved Best Practices, and Certification Programs that, when implemented, will improve the quality and integrity of the lottery environment, and will provide increased efficiencies, resulting in reduced costs and increased profit margins for lotteries, vendors, and lottery retailers.

The NSI mission is to establish a resilient organizational structure, set of processes, and procedures that will engage all constituents (lotteries, vendors, and retail representatives) in an environment of open discussion and cooperative development.

The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX certification.

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, the main part of which is focused on development of Technical and Product Standards, Best Practices, and Guides. Full details and a catalog are available at www.opengroup.org/pubs.

This Document

This document is the Best Practice for Quality Assurance of Product Development in the Lottery Industry, specifically addressing the Development Process. It has been developed and approved by NASPL in association with The Open Group.

Trademarks

Acknowledgements

NASPL and The Open Group gratefully acknowledge the contribution of the following people in the development of this Best Practice:

James Andrews	The Open Group
Robert Andriola	Scientific Games International
Richard L. Chavis Sr.	Maryland State Lottery Association
Patt Eberhart	California Lottery
Karen Fournet	Louisiana Lottery Corporation
Cathy Fox	The Open Group
Patrick Frament	New York Lottery
Margaret Gibbs	Kentucky Lottery Corporation
Caroline Ginglen	Scientific Games International
Shawn Hawley	Scientific Games International
Daniel Johnson	Georgia Lottery Corporation
Kumar Kalagara	California Lottery
Bob Little	Kentucky Lottery Corporation
Sally Long	The Open Group
David MacDonald	GTECH Corporation
Marc Rene	GTECH Corporation
Harvey Roberts	Kentucky Lottery Corporation
Deborah May Schoonover	The Open Group

1 Introduction

A Best Practice provides a clear description of a set of processes, procedures, and guidelines, that when practically applied to an operation brings a business advantage. A Best Practice has a record of success in providing significant advantage in cost, schedule, quality, integrity, performance, safety, environment, or other measurable factors that impact an organization. Various organizations identify and publicize Best Practices so that others, particularly internal business units, external business partners, or otherwise affiliated external organizations, can benefit from implementing the Best Practice and improving the operation of their business.

Best Practices can be applied to particular subject areas (such as new technologies or management theories), product sectors (such as software and hardware development), and vertical markets (such as the lottery industry). Best Practices are used frequently in the fields of healthcare, government administration, education, project management, hardware and software product development, and elsewhere. A commitment to using the Best Practice in any field is a commitment to using a prescribed method to ensure success.

A NASPL Best Practice is a Best Practice that applies to the lottery industry, has been approved by the NASPL Standards Initiative (NSI), and which serves as a recommendation for adoption by the lottery industry. A NASPL Best Practice is a practice that when implemented is intended to improve the quality and integrity of the lottery environment, and to provide increased efficiencies, resulting in reduced costs and increased profit margins for lotteries, vendors, and lottery retailers.

A NASPL Best Practice is described in terms of its:

- Purpose
- Components
- Constituents and their roles
- Prescriptive requirements
- Methods and techniques
- Tools
- Relationship to other Best Practices

The development of a NASPL Best Practice involves the following stages:

1. The NSI, through the Best Practices Working Group, selects a candidate practice using specific assessment and acceptance criteria (as defined by the NASPL Steering Committee).
2. The Best Practices Working Group develops a Best Practice document.

3. Optionally, the Best Practice document is subject to an informal review process by NASPL members and the NSI participants.

4. The Best Practice document is subject to a formal review process by the NSI Steering Committee and the Best Practice Review Board.

5. A set of conformance criteria and a conformance policy for the Best Practice are defined.

The approved NASPL Best Practice describes the practice in enough detail to enable it to be readily deployed by other organizations, assuming the availability of the necessary resources.

This chapter describes this NASPL Best Practice in terms of its purpose and its scope, and gives a definition of the terminology used throughout this document.

1.1 Purpose

This Best Practice is one of three addressing Quality Assurance of Product Development in the Lottery Industry:

1. Requirements Definition (Doc. No. BP0401)

2. Development Process (this document)

3. Acceptance Testing (Doc. No. BP0403)

The purpose of this set of Best Practices is to provide a documented set of quality assurance processes and procedures that will allow lotteries and vendors to follow an approved and repeatable method for the purpose of meeting the goals and objectives of the lottery through hardware and software deployment.

1.2 Scope

This Best Practice – together with the associated Requirements Definition and Acceptance Testing Best Practices – provides a set of processes and procedures that address the quality assurance requirements throughout the hardware and/or software production cycle from requirements specification through design, implementation, and testing, to acceptance and deployment. The scope of this Best Practice, although general enough for many software and hardware production environments, has several quality assurance aspects that are specific to the lottery industry. This Best Practice is not intended to cover procurement of off-the-shelf applications or ready solutions.

This set of quality assurance Best Practices is related in some ways to the Request For Proposal (RFP) process, particularly with respect to Requirements Definition. The RFP process itself is the subject of another NASPL Best Practice, and is outside the scope of this document.

1.3 Terminology

This section provides a set of terms and their definitions, which should be used when describing and interpreting the Best Practice requirements of the quality assurance processes and procedures specified in this document.

Must	Indicates an absolute, mandatory requirement of the Best Practice that has to be implemented in order to conform to the Best Practice.
Should	Indicates a recommendation that ordinarily must be implemented. To conform to the Best Practice, an acceptable justification must be presented if the requirement is not satisfied.
May	Indicates an optional requirement to be implemented at the discretion of the practitioner, and which has no impact on conformance to the Best Practice.
Must not	Indicates an absolute preclusion of the Best Practice, and if implemented would represent a non-conformity with the Best Practice.
Should not	Indicates a practice explicitly recommended not to be implemented. To conform to the Best Practice, an acceptable justification must be presented if the requirement is implemented.

2 Best Practice Environment

This section describes the typical business environment, the business drivers, and the objectives driving this NASPL Best Practice as context.

2.1 Business Environment Summary

2.1.1 Business Scenario – General Description

This section describes the stakeholders in a typical lottery operation. The roles played by the constituents are not necessarily the same for every lottery. The constituents may take on different roles during the execution of business processes based upon local practice, how the lottery is organized, the budget allocated to the lottery organization, or any number of other factors. Therefore, in one jurisdiction a constituent may take a role that is taken by another constituent in another jurisdiction; for example, developing software or hardware for a lottery may be done by a vendor or by a lottery organization. These roles may actually change over time.

The key organizations and entities in the typical lottery business environment – particularly those relevant to the processes discussed in this Best Practice – are illustrated in the following figure.

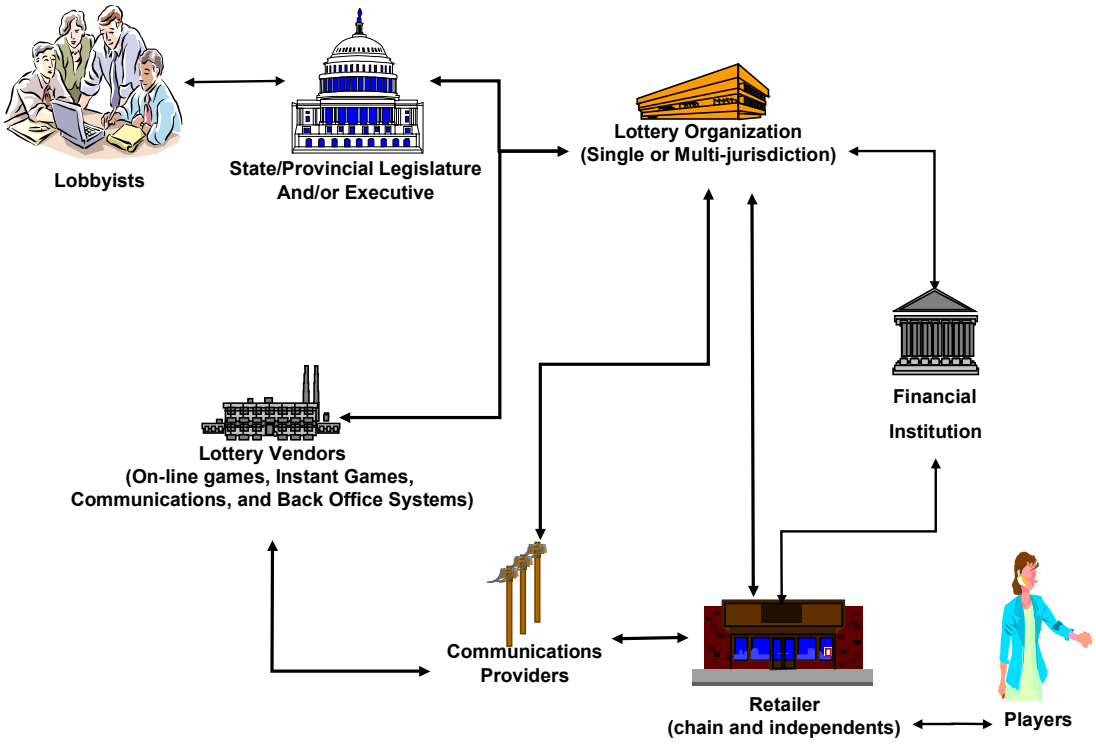

Figure 1: The Lottery Business Environment

Not all organizations will have all of these components and relationships. However, the figure illustrates a number of points typical of lottery enterprises, each of which has particular implications for the benefits of standards for the lottery industry. Points to note regarding some of these constituents and their relationships to this Best Practice:

- Lottery vendors provide solutions for many lotteries; therefore, a reduction in the need for per-lottery customizations and variable quality assurance requirements improves the efficiency of the lottery vendor.
- Lotteries may buy different systems or components from different vendors over time, so developing a set of quality assurance Best Practices that are understood and followed by all vendors and lotteries will result in less time required in defining the practices and procedures to be followed during the product development lifecycle.
- Many lotteries participate in multi-jurisdiction games, bringing additional (and shared) governance processes and operational requirements. If requirements for quality assurance are defined as standard best practice among all lotteries, there would be less time and effort spent in meeting the cross-jurisdictional requirements, once a lottery decides to participate in a multi-jurisdiction game.

- The lottery retailers serve the players, and must do so in a timely fashion as most of the retailers are convenience (or sometimes grocery) stores where even occasional long waits resulting from poor quality gaming software or hardware systems can have significant impact on revenue.

The following list of constituents and the roles they play in the larger lottery environment is provided here to give a big picture view. The constituents involved in this Best Practice, and the roles they play are a subset of those in the larger lottery environment and are identified in more detail in subsequent sections.

Constituent	**Role Played**
State Executive or Legislature	Authorize lottery operation under state/provincial laws. Direct use of lottery revenues (and by implication, lottery operating budgets). Monitor and audit lottery operations, sometimes impacting lottery development. May appoint lottery director.
Board of Directors / Lottery Commissioners	Oversee lottery organization and their policies and procedures. Hire lottery executives. Approve major lottery contracts.
Lottery Organizations	Conduct overall operation of the lottery. May operate lottery IT infrastructure. May develop games. Oversee lottery integrity and security, including validation of winners. Optimize profitability from games (current and future), selecting new games, stopping old games, developing new games, and managing the selection and implementation of game infrastructure through Requests For Proposals (RFP). Manage retailers; including accounting, and game material inventory; e.g., instant game books. Manage vendors, including possible outsourcing of lottery operations. Develop marketing campaign. Manage large prize payouts individually or in conjunction with multi-state organizations.
Retailers/Agents	Sell lottery tickets and games at retail location. Market lottery products. Validate and redeem tickets. Manage and account to lottery for sales including ticket “books”, report sales to lottery commission, redemption of unsold game books. Manage accounting of lottery contribution to store profit and loss. Optimize contribution of lottery sales (within lottery regulations) to store.
Financial Institutions (e.g., banks)	Provide “sweep accounts” to facilitate transfers of funds from on-line and instant ticket purchase between the retailer/agent and the lottery. May provide interface between state treasury and lottery.
Players	Play on-line and instant games, self-validate tickets (in some jurisdictions), redeem tickets, and receive winnings.

Constituent	Role Played
Lottery System Vendors	Provide lottery systems, components, games, and/or products. May provide the networking component (possibly customized) of a lottery system. Operate lottery IT systems (under subcontract from lottery organization) in many jurisdictions. Provide maintenance, field, and technical service in some jurisdictions. Respond to Requests For Information (RFI), Requests For Proposals (RFP), and Requests For Software Changes (RFS).
Telecommunications Providers	Provide the networking component (possibly customized) of a lottery system.
Lobbyists	Impact lottery responsibilities and limitations (through legislature) within a jurisdiction.

2.1.2 Operational Scenario

To provide an understanding of where and when the quality assurance Best Practices will be applied, this section depicts a typical product development lifecycle as it applies to the lottery industry: from specifying requirements to acceptance testing. This operational scenario highlights the major processes and illustrates the associated need for quality assurance Best Practices. It also identifies the constituents who will be carrying out the quality assurance Best Practices.

The software or hardware development cycle and the quality assurance processes associated with it begin with articulating requirements for new or upgraded software or hardware, which is done primarily by the appropriate lottery personnel within their own environment. Once the requirements are documented, they are handed off to the vendor. After the requirements are agreed, the vendor begins their development cycle incorporating quality assurance processes that cover design, implementation, testing, and release procedures. When the product is "released" to the lottery, the lottery, with the vendor supporting the process, performs acceptance testing to determine whether the product meets the acceptance criteria typically defined initially during Requirements Definition.

The constituents involved in this Best Practice are as follows, with the lotteries and vendors having a major role in most of the processes, and the retailers having a minor role in some jurisdictions:

- Lotteries
- Vendors
- Retailers/Agents

The operational environment for this Best Practice is:

- **Dynamic** – Lotteries continually upgrade existing games and institute new games so that their business can evolve and grow. High availability with optimum performance and quality software and hardware are essential in the lottery business so that downtime during upgrades, deployment of new games, and ongoing operations must be minimal.

- **Diverse** – Since there is no enforcement of a common method among lotteries, every jurisdiction's operation executes slightly differently and according to its own method of choice and interpretation. However, there should be an effort to provide commonality between states where possible.
- **Local and culturally-specific** – Geographical differences mean that jurisdictions vary, manifesting in diverse needs. This represents diversity in participants and method, including cultural differences.

It is imperative that the quality assurance Best Practices support this business environment.

2.1.2.1 Operational Functions and Processes

The key functions and processes that require best practice support are further identified in the table below. The specific needs within each business function or process requiring best practice support are also described.

Function/Process Name	Best Practice Needs
Requirements Definition	The requirements for the system or system components must be defined, documented, agreed, and approved by both the supplier and customer of the system. Best Practices need to validate that this process happens, that the correct information is specified, and that the right processes are used in reaching agreement between vendors and the lottery.
Development Process	Use of a development process that covers design, implementation, testing, problem tracking and resolution, change control management, and release and installation. The process needs to include adequate documentation and approval phases.
Acceptance Testing	Utilize a defined Acceptance Testing process and plan that is typically agreed during Requirements Definition and is carried out in a controlled environment during Acceptance Test Execution.

2.1.2.2 Operational Topology

The topology of the environment to which this Best Practice applies (below) typically represents distributed and separate locations with variable overlap, and sometimes complete overlap, between some of these entities:

- Lottery Organization
- Retailer Site
- System Vendor
- ICS Vendor

2.1.2.3 Operational Location Information

The following matrix shows the *primary* locations where each of the functions or processes related to this Best Practice is executed, though all identified locations may not be involved in every situation. In cases where different parts of a function or process involve different locations, the component parts of the function or process are identified. This demonstrates the need for integration of different requirements when creating and adopting this Best Practice.

Functions/ Processes	**Locations**			
	Lottery Organization	**Retail Site**	**System Vendor**	**ICS Vendor**
Requirements Definition	X	X	X	
Development Process				
– Software and Hardware Design			X	X
– Software and Hardware Implementation			X	X
– Vendor Internal Testing			X	X
– Acceptance Test Readiness Review	X		X	
– Change Control System			X	
– Problem Reporting	X	X	X	X
– Problem Resolution			X	X
– Release Process	X		X	
Acceptance Testing	X		X	X

2.2 Business Rationale

This section describes the business drivers, objectives, and benefits of implementing this NASPL Best Practice.

2.2.1 Business Drivers

The major business drivers for implementing a Best Practice for Quality Assurance of Product Development in the Lottery Industry are the potential for reduced risk and increased integrity for the lotteries, reduction in development costs, decreased potential for lost revenue, and decrease in rate of project failure. These business drivers are summarized below:

- **Liability**

  The potential for liability issues exists if faulty software or hardware is installed. The types of conceivable issues depend on the type of software or hardware being installed, but issues could include large dollar liabilities in the case of incorrect tickets being generated or paid. Even with the protection of rules and regulations, which attempt to limit the liability issue, legal issues will still arise and may ultimately be successful. Preventing problems by following Best Practices for quality assurance before deployment will help prevent the costs after the fact.

- **Lost revenue**

  This business driver is associated with the costs to the business in terms of lost sales or productivity when the system or supporting networks are down or performance is poor, in the retail environment or at the lottery central office.

- **High costs associated with fixing problems in the field**

  Development time and costs are decreased if problems are discovered and resolved during testing in the development or acceptance cycle rather than after installation and deployment in the field.

- **Public relations and loss of integrity**

  Public relations problems can result from the installation of defective software or hardware systems in the lottery or retailer environment. Any problem a lottery incurs that becomes public has the potential for negative consequences and negative publicity, which can ultimately turn into concern on the part of the public about the integrity of the lottery.

- **Loss of initiative**

  When a new program initiative fails, the impact often extends beyond that of the current initiative. For example, if the software implementation of a game change goes badly, a lottery may be reluctant to run other games of a similar type or to introduce other new types of games. This can result in a lottery having a lower risk tolerance for introducing other innovative programs in the future.

2.2.2 Objectives and Benefits

This section outlines some of the business objectives for introducing the quality assurance Best Practices and some of the benefits that could be attained once these Best Practices have been adopted.

The major objective for implementing a Best Practice for Quality Assurance of Product Development in the Lottery Industry is to increase the integrity of the lottery business by producing higher-quality systems, which could result in the following benefits:

- **Reduced risk associated with liability** by providing assurances that the systems have been through approved quality assurance Best Practices, particularly during the Development Process and Acceptance Testing.

- **Reduction in lost revenue** as a result of better performance and higher availability.

- **Reduced costs** associated with development, testing, and maintenance.
- **Increased integrity,** resulting in greater public confidence and player satisfaction that could lead to increased sales.

3 Best Practice Overview

This chapter provides an overview of the Development Process Best Practice.

To assure quality in lottery systems, it is considered best practice to follow a consistent set of processes and procedures in three distinct phases of the product development lifecycle of a lottery system as follows:

- Requirements Definition
- Development Process
- Acceptance Testing

This Best Practice addresses the Development Process.

Development is the process by which the vendor produces the product to be delivered to the lottery. This Best Practice focuses on the quality engineering practices that the vendor employs to assure that the product is produced in accordance with the requirements, timeframes, and quality criteria defined and agreed during Requirements Definition.

The applicability of this Best Practice extends to all areas of software and/or hardware production for the lottery industry, including:

- Production of a new lottery system
- Creation of new software and/or hardware components for use in an existing lottery environment
- Updates or extensions to existing lottery system components

3.1 Relationship with Other Best Practices

This Best Practice is one of three that address Quality Assurance of Product Development in the Lottery Industry, as follows:

- Requirements Definition (Doc. No. BP0401)
- Development Process (this document)
- Acceptance Testing (Doc. No. BP0403)

This set of quality assurance Best Practices is related in some ways to the Request For Proposal (RFP) Best Practice, particularly with respect to Requirements Definition. The RFP Best Practice is centered around publication of the RFP, submission and evaluation of the response, and vendor selection, whereas the quality assurance Best Practices are focused on Best Practices

that should be implemented after the vendor has been selected by the lottery to provide the product(s) or upgrade(s).

4 Best Practice Description

This chapter defines the Best Practice for the Development Process used by the vendor. Though this Best Practice primarily applies to and is written in reference to the vendors who are creating IT product for the lotteries, it should be noted that in the context of this Best Practice, the term "vendor" applies to any constituent in the role of product developer and in particular can be applied to lotteries who are engaged in in-house development of lottery technology and applications, and to third-party application vendors.

4.1 Overview

Development is the process by which the vendor produces the product to be delivered to the lottery. In order to assure the quality of the resultant product and assure that the requirements have been correctly implemented, the vendor will utilize various processes and procedures during the product development lifecycle. The objective of this Best Practice is to identify the development processes and procedures that constitute good quality engineering practice, and establish best practices for how these processes and procedures are deployed during the product development lifecycle.

To apply this Best Practice, a vendor will need to have development processes and procedures that are being used within their organization, have them documented so they are able to be deployed across the relevant parts of the organization, and have established mechanisms to ensure that development activities are performed in accordance with the organization's defined and documented processes.

The purpose of this Best Practice is not to suggest that a vendor use a specific development model or set of tools, but rather to utilize their chosen model and tools in accordance with this Best Practice.

Quality engineering practices typically involve the use of defined hardware and/or software development methods. Likewise, good project and quality management are fundamental to achieving time, cost, and quality goals. The vendor's documented processes and procedures will likely span the full product development spectrum, covering requirements definition and evolution, project management and progress monitoring, design, implementation, testing, internal and external acceptance, problem tracking and resolution, change control, release, and installation. The process will likely include technical review of documents, software and hardware deliverables, and project phase reviews at appropriate points in the product development lifecycle.

This Best Practice addresses the following major processes in the product development lifecycle and specifies best practice requirements for each of them in subsequent sections:

- Detailed Design
- Implementation

- Internal Testing
- Acceptance Test Readiness Review
- Release Process
- Change Control Management
- Problem Reporting

4.2 Constituents and Roles

The vendor organization that is producing a product for a lottery is the primary constituent in the Development Process. There are many parts of the vendor organization that may be involved in the Development Process, including but not limited to:

- Software Engineering
- Quality Assurance
- Release Engineering
- Integration Testing
- Project Management

The role of the lottery in the Development Process is in the Acceptance Test Readiness Review process, to work with the vendor in determining when the product is ready to transition from the Development Process to Acceptance Testing.

4.3 Components

4.3.1 Detailed Design

Detailed design is the process by which requirements are decomposed into software or hardware modules, and design decisions are documented. The detailed design process provides the basis for implementation and testing. For example, the types of design that could be performed during the detailed design stage include the design of the data dictionary, detailed file structures, module logic descriptions, and pseudo-code algorithm definitions.

4.3.1.1 Best Practice Requirements

The vendor must have a documented process for performing detailed design that should include all of the following:

- Standards for the format and content of the design description
- Traceability to the system design description and requirements
- Technical review of the detailed design

- Procedures for tracking and managing changes to the design throughout the product development lifecycle (see Section 4.3.6 (Change Control Management) for further information)
- Procedures for ensuring that design constraints or other areas of risk have been considered and assessed

Detailed design must be performed in accordance with the vendor's documented design process.

4.3.2 Implementation

Implementation is the process of producing the hardware and/or software modules necessary to realize the requirements. For software, it is the creation and/or integration of code modules. For hardware, it is the prototyping and fabrication of the hardware modules.

4.3.2.1 Best Practice Requirements

The vendor's approach to implementation must be documented.

Any processes or technical criteria used to support the implementation process, such as checklists, should be formally documented.

For software component implementation, the software development practices employed should include one or more of the following practices:

- Coding is done in accordance with the vendor's documented coding standards.
- Code walkthroughs, inspections, or reviews are performed.
- Formal methods are used as a software development method to establish the correctness of the implementation of each requirement.
- Other specified methods are used to ensure the correctness of the implementation.

4.3.3 Internal Testing

Internal testing is the vendor's process of internal validation of all the project deliverables, both as individual components and as a whole in an integrated system. The latter part of this process is often known as system validation.

4.3.3.1 Best Practice Requirements

A vendor's internal testing must be performed according to a defined testing process.

The defined testing process should include:

- Test planning
- Test specification
- Test execution
- Test reporting

The defined testing process may include:

- Root cause analysis

The internal testing process should be based on, and closely tied to, the specifications for the system and its various components.

The internal testing process must be documented in a vendor's internal test plan.

Internal testing must be executed in accordance with this test plan.

The vendor's internal test plan must define the method and processes that the vendor will employ during internal testing to assure that the requirements have been correctly implemented and that the system is suitable for delivery to the lottery.

The vendor's internal test plan must cover:

- How the vendor intends to test each new component, and each significant software or hardware change
- All levels of testing to be performed as appropriate for the scope of the deliverables; this should include, but is not limited to:
  - — Unit testing
  - — Integration testing
  - — System testing
  - — Regression testing
- For each level of testing, the specific types of testing to be performed; Appendix C defines the various types of testing options that may be employed during system testing
- Exit criteria that must be fulfilled in order for the vendor to declare the system components suitable for delivery to the lottery
- Test reporting that summarizes the results of testing must include:
  - — Any deviation between observed and expected behavior of the system components
  - — Any defects identified

Final system testing should not be performed by the same people who designed the hardware or wrote the software being tested.

4.3.4 Acceptance Test Readiness Review

This review is performed with representatives of the vendor and lottery to determine whether the product is ready to transition from the vendor's Development Process to Acceptance Testing. During this review, vendor and lottery representatives will review the handover criteria and other relevant materials in relation to the current state of the product. It is the responsibility of the vendor to schedule this meeting with the lottery.

4.3.4.1 Best Practice Requirements

Representatives from the vendor and lottery organizations must hold a review meeting to determine whether the product is ready to transition to Acceptance Testing.

The acceptance test readiness review meeting must include the following:

- Vendor's test summary report, which must include:
  - — A summary of the types of testing performed
  - — A summary of the results of that testing
- Open defects and issues
- Lottery's Acceptance Test Plan, which must cover:
  - — Whether the lottery's entry criteria for Acceptance Testing have been fulfilled
  - — Training readiness of the lottery with respect to installation and operation of the system
  - — What the responsibilities of each party are during Acceptance Testing
- A definition of the acceptance test environment, which must define exactly what is required to conduct Acceptance Testing

At the conclusion of the readiness review, either the lottery must authorize the handover for Acceptance Testing, or the vendor and lottery must agree what steps are required for the product to be approved for Acceptance Testing.

4.3.5 Release Process

The release process is the process by which the internally accepted system and its associated documentation deliverables are prepared for delivery to the lottery for its acceptance. Once delivered, the new or updated product will need to be installed and integrated into an existing lottery environment.

The release process covers both the preparatory steps performed on the vendor side, and the documentation necessary to facilitate the installation and integration performed on the lottery side. The release process addresses installation plans, start-up strategies and plans, and conversion requirements. The release process could also include training of operations staff, the lottery, and retailers.

4.3.5.1 Best Practice Requirements

The plans for the release of the system or system components must be documented, and the documentation should address the following:

- Means of delivery
- Delivery medium
- Lottery acceptance process and acceptance test period

- Knowledge transfer from the vendor to the lottery regarding the use and operation of the system
- Problem reporting mechanism to be used after delivery; that is, during Acceptance Testing and post-installation (integration). In some lottery domains, the problem report is referred to as a Testing Incident Report (TIR).
- Release notes

The release notes should include at a minimum:

- A formal build description for the software portion of the release, which must include:
  - — The system version
  - — The version of all component modules
  - — The version of all deliverable documents
- A list of any known problems in the release and the plans for resolving them
- Any limitations in the scope of the system, or any requirements that were included in the specifications but were deferred or eliminated and will not be implemented as a part of this release
- A definition of any additional functionality that is being delivered but was not defined in the Requirements Specification (i.e., enhancements); this must include a reference to where the functionality is documented
- A list of all deliverables, which must include:
  - — A description of and identifying serial and part numbers for all hardware components
  - — Software components and associated software modules
  - — Documents
  - — Tools
- Test report that summarizes the test results of internal testing
- Any known usability constraints (for example, performance limitations)
- Any known portability constraints
- Any known interoperability constraints

The lottery and the vendor must work together to produce an integration plan. The lottery and vendor should produce the integration plan during Requirements Definition.

The integration plan should account for:

- Reviewing the business processes impacted by the changes
- Ensuring that people responsible for the affected business functions know of the changes

- Ensuring that the people responsible for the affected business functions are notified in advance of what business process changes need to be made (if any)
- Ensuring that these non-software issues are resolved in time to support the installation of the new software or hardware

The integration plan must include information that covers the complete environment into which the product will be integrated and deployed. This information must identify the impact of the product release on the overall lottery system, including but not limited to the following:

- If any security changes are required, the security changes must be identified in the integration plan together with assigned actions for the responsible parties from either the vendor or the lottery.
- If any data conversion is necessary the vendor must provide details and assigned actions for execution of the data conversion.
- Identification of platform variations that may occur at different lottery or retail sites and document assigned actions to address them.
- Identification of deployment variations that may occur at different lottery or retail and document assigned actions to address them.

The integration plan must be approved by both the lottery and vendor.

The integration plan may be refined during Acceptance Testing.

Integration must be performed in accordance with the agreed plan.

For integration purposes the vendor must:

- Provide an installation guide that is unambiguous and includes installation instructions as well as instructions for backup and back-out procedures where appropriate.
- Thoroughly test the installation procedures defined in the installation guide prior to release.

4.3.6 Change Control Management

Good quality engineering practices depend on a defined change control management process that includes mechanisms for tracking the progress of change requests to system features and modifications to code and documentation. It also enforces the association between the change requests and the code and documentation modifications.

An effective change control management process also covers the mechanisms for scheduled and controlled insertion of new code into the current configuration. Scheduled and controlled changes mean that the vendor and the lottery agree on when and how changes will be applied, and mechanisms are in place to prevent and detect unplanned changes.

A well-designed change control process enables vendors and lotteries to evaluate the time and cost impacts of a change request prior to committing to make the change. It also includes mechanisms to ensure that each change made is exactly what was requested and that it was

properly specified, coded, tested, and integrated into the system. It is good practice for the lottery to independently test the vendor's delivery of a change prior to installation of the change.

4.3.6.1 Best Practice Requirements

The vendor must have a documented change control management process that defines the requirements and process for making modifications to items under change control.

The vendor's documented change control management process should address all the following aspects of change and configuration control:

- Identification of the items under change control
- The relationship between problem tracking and change control
- Mechanisms for change control (e.g., most software development projects use an automated system such as CVS)
- Document modification (in particular, it is essential that modifications to requirements are tracked)
- Build status tracking (i.e., the relationship between the module version and the system version)
- Contract modification (a special case of change control requiring formal approval by both the lottery and vendor)
- Criteria for approval of different change control elements

The change control management process should include procedures for submitting change requests and tracking the progress of a change request.

A change request must include the following:

- A unique tracking number provided by the change management system
- Description of the change requested
- Reason for the request
- Date of the request
- Who requested the change
- Component or system which is effected

The change request should pass through defined stages as it is being addressed.

The change request tracking mechanism must provide all relevant parties with access to the status of defects which:

- In the case of products under development, are adversely affecting milestone delivery dates

  or:

- In the case of products in the field, are adversely affecting operations

When a change request is approved that requires software modification, the change control process should require use of a software and documentation modification tracking mechanism.

The change control tracking mechanism must require certain information including:

- Module
- Version
- Code modification linked to the version

The change control tracking mechanism may include additional information including:

- Check-out date
- Engineer
- Size
- Check-in date

The change control process should require that code modifications be versioned and documented before they are checked back into a library.

If a source code modification alters the system functionality, the change control process should require that the system functional specifications be updated and versioned.

4.3.7 Problem Reporting Process

Good quality engineering practices depend on a defined problem reporting process. The problem reporting process provides a mechanism to track perceived system defects or issues. It is typically the vendor who will administer this mechanism, though that is not a requirement. A well-designed problem reporting mechanism will:

- Accurately record all issues that are reported from vendors, lotteries, or lottery-sponsored auditors
- Define the stages that a problem report will pass through, document progress through those stages, and restrict the ability to skip stages
- Define how problem reports are prioritized and how resolution of the issue will be managed
- Make the documentation of the progress available to all interested parties and provide a current status for every issue

4.3.7.1 Best Practice Requirements

The problem reporting process must provide a mechanism that allows each issue to be submitted in a standardized format with standard content, and provide all relevant parties with access to the most current data.

Each issue must have a unique tracking identifier.

Submission of an issue must include the following information:

- Description of the issue
- Component or system in which the issue was found
- Severity of the problem
- Date the issue was submitted
- The submitter of the issue
- Process in which the issue was reported, such as vendor internal testing, acceptance testing, or production
- Product version in which the issue was found

Information included in the submission of an issue may include:

- Suggestion of how to recreate the issue; for example, pay a ticket after the pay-out period expiration

The problem reporting process must include a defined process for reviewing and resolving problems. A team of representatives from each key party should perform the review of problem reports. This team may include representatives from:

- The lottery quality assurance team
- The development team
- The project management

Each problem submission should be reviewed by the review team to determine whether it requires immediate action or if it may be deferred to a future release. The review should consider all of the following:

- The overall risk associated with fixing the problem (this may increase or decrease risk)
- The cost associated with fixing the problem
- The impact on the lottery system or system component associated with fixing the problem

The problem reporting process should include mechanisms that prevent any one individual or group from moving an issue from initial submission to closure. To facilitate this, a problem report should flow through distinct, defined stages.

At the completion of each stage, the responsible individual should update the problem report with the action taken and a time and date stamp. This ensures that the current status of any issue is available at any time.

The problem reporting mechanism should provide a means to track issues and their resolution to ensure that critical problems found during one stage are fixed and integrated prior to delivery to the next phase.

The problem reporting process must ensure that the fixes are integrated following the change management system guidelines.

5 Methods, Techniques, and Additional Considerations

This section describes methods and techniques that support the Development Process.

5.1 Detailed Design

There are currently no methods or techniques defined in this Best Practice for the detailed design process. However, the OMG's Model Driven Architecture (MDA)[1] is a development method that is showing great promise for the development of software-intensive systems.

5.2 Implementation

5.2.1 Maintainability

Good system development standards are devised with maintainability in mind.

5.3 Internal Testing

5.3.1 Test Planning

Internal testing is based upon a documented test strategy and plan. The test plan documents the underlying risks and assumptions that were taken into account when creating the plan. The test plan clearly describes how critical delivery risks are mitigated and how new risks, if identified, will impact the existing plan. The plan identifies the specific types of testing (refer to Appendix C) and the rationale for the selection of specific testing types.

5.3.2 Exit Criteria

Internal testing exit criteria form a contract between the internal development team and the internal test team. Exit criteria define the minimum set of functionality to be delivered as well as the maximum amount of defects that can exist for the product to progress to the next phase. In addition, exit criteria can be used for any other conditions that are required for the delivery to be successful. It is good practice to document exit criteria for each internal test phase documented in the test plan.

[1] Model Driven Architecture® is a registered trademark of the Object Management Group, Inc. in the United States and/or other countries.

5.4 Acceptance Test Readiness Review

To ensure the readiness review meeting is productive, it is good practice to distribute the materials to be reviewed at the meeting to all of the review participants. The materials to be reviewed include the vendor's test summary report, the lottery's Acceptance Test Plan, the list of open defects and issues, and the definition of the acceptance test environment. These materials should be distributed well in advance of the meeting, to allow sufficient time for them to be read and understood.

One key objective of the review of the Acceptance Test Plan is to set expectations on what will happen during acceptance testing and what is expected of each party. Any test documents to be used by the lottery in addition to the Acceptance Test Plan also need to be included in the review materials.

When reviewing the acceptance test environment, things to look for are whether the environment is set up and ready-to-use, and whether it meets its intent. If the environment is not yet set up, it is a good idea to discuss the steps required to set up and maintain the environment and agree who is going to perform the various steps. The intent of the environment set-up is also taken into account; for instance, if the environment is intended to mimic the production environment, the review covers what is required to mimic that environment and evaluate whether the test environment actually achieves that.

5.5 Release Process

The use of checksums allows for software delivery and installation to be confirmed. A checksum is an identifier for a file or a set of files created by viewing the file(s) as a series of data items (usually bits) that, when combined uniquely, identify the file(s) and allow for the content and structure to be verified. Checksum algorithms are written so that any change to the file will produce a different checksum and that it is extremely unlikely that two different files or set of files will have the same checksum.

Checksums (or similar mechanisms) serve several purposes:

- They are used to verify the correct installation of the release.
- They are used to detect media errors (i.e., bad tape or sectors on a disk).
- They are used to prevent fraud by assuring what was built by the development team is what is installed on the destination system.

Before a software release, a checksum calculation is performed and the value is noted in the release notes. Once the release is installed, a checksum is computed on the destination system and the two checksums compared. If any difference is found, an investigation occurs to determine and correct the cause of the difference.

Through the use of checksums, it is possible to verify that executables in the production environment match those from a system build. In addition, it is possible to identify the source code files that were used to build the system. Correct and consistent use of checksums will significantly reduce the risk of liability due to unwarranted or incorrect code installed in the production environment.

5.6 Change Control Management

It is good practice for the lottery and vendor to use a change control management process encompassing an implementation team consisting of representatives of both the lottery and vendor, as well as the vendor's project manager or designee. The vendor's representative in this context applies approved changes and reports to the implementation team that the approved changes have been made, and also that no unapproved changes have been made.

A change control management process may be implemented using electronic techniques, or it may be manual. If the mechanism is electronic, access to the change request information in the system may be provided through a web page, database query, or a report. If the mechanism used is manual, it is a good idea to send the change request information to all relevant parties on a scheduled basis (such as weekly). This can be done effectively by sending a report via email.

An effective change control management process contains criteria for approval of different change control elements. A good example of such criteria is:

- Editorial changes to documents only need the approval of the document author.
- Changes that may affect time, cost, or quality need a formal written change control procedure.
- Technical changes to code require more than one person’s approval.

An effective change control management method defines several stages that a change request flows through as it is being addressed. Examples of the stages a change request may go through are:

- Submission
- Specification
- Sizing
- Approval
- Implementation
- Closure

Each stage has a defined input and an output that explains the tasks performed in the stage. Inputs and outputs in each stage may include:

- Submission stage:
  In: Request for a change
  Out: The change request – approval or disapproval of the change request
- Specification stage:
  In: The change request
  Out: The signed, approved requirements document

- Sizing stage:

  In: The signed, approved requirements document

  Out: The hours of work and capital costs associated with the request

- Approval stage:

  In: The signed, approved requirements document and the costs associated with the request

  Out: The written approval or disapproval to proceed

- Implementation stage:

  In: The approved requirements document

  Out: The tested, modified software/hardware feature and the related documentation

- Closure stage:

  In: The modified software/hardware features

  Out: Proof that the change has been made available or delivered

A software and documentation modification tracking mechanism may be used for tracking software modifications. There are automated version control applications in common use such as SCCS, CVS, or ClearCase that would fulfill this requirement.

An effective change control management process may require that code modifications be versioned. This is good practice because versioning allows:

- The capability to reproduce the software at various milestones, and allow for patches to be created against those releases
- The capability to identify, and if necessary, reverse any source code modifications from the code base
- Parallel development in cases where multiple deliverables need to coexist in a single code base without interfering with each other
- Tracking the build tools, third-party libraries, and operating systems that make up the environment in which the product is manufactured

A good method for detecting any changes (expected or unexpected) to the production software configuration is to produce a checksum on the executables. This may be done each day when the lottery system initially boots up. If a comparison of the checksum to the prior day's checksum indicates a difference, this means that a change was made to the production software configuration. If a change had been planned, then this method validates that the change was made. However, if a change was not intended, then this method indicates a problem that needs to be investigated.

5.7 Problem Reporting Process

A problem reporting process may be implemented manually, or using an electronic mechanism, such as a website, spreadsheet, or database. Regardless of the implementation mechanism, it is

important to ensure that all steps in the problem reporting process are supported and that all relevant information regarding a problem report is captured, stored, and managed.

There are several distinct stages through which a problem report flows. Examples are:

- Submission – This is the initial stage for a problem report.
- Development – This stage is for when a problem is being fixed by development. This may include sub-classification for the particular phase within development that the problem is currently in, such as design, implementation, or unit test.
- Test – This stage is for when a problem fix is being tested by the lottery. This may be a sub-classification for the particular aspect of the testing process that the problem is currently in, such as checklists, or test scripts.
- Closure – This is the final stage of a problem report and includes a classification for the type of resolution, such as whether the problem was fixed or dropped.

When a problem report moves from one stage to another, it is good practice to update the problem report to include information on the actions taken and relevant information for the next stage. Examples of the types of information that may be useful to include at the end of each stage are as follows:

- Submission stage:
  - — Was the issue a software defect, a set-up issue, or a misunderstanding of the system functionality?
  - — Was the issue able to be reproduced, and how?
- Development stage:
  - — What is the design to solve the issue?
  - — Was the code correctly modified to address the issue, and how was that fact determined (e.g., code review, peer review, supervisor approval)?
  - — Did the software perform properly in unit test, and how was that determined (e.g., peer review, supervisor approval)?
- Test stage:
  - — Did the test include a checklist designed to ensure that the problem will not occur again?
  - — Did the test use test scripts to ensure that the fix did not corrupt some other part of the system (e.g., regression tests)?
- Closure stage:
  - — How was the issue resolved? Or why was the issue dropped?
  - — If resolved, what is the version and what is the date when the modification is scheduled to ship?

It is good practice for the vendor or lottery to attempt to classify or sort problems by type or component as part of a quality improvement process. For example, if 60% of problems arise from faulty requirements definitions, then that suggests a change to the underlying Requirements Definition process. If 60% of the system defects arise from Module X, then perhaps Module X needs a special review.

6 Tools to Support the Development Process

At this time we do not list specific tools, but rather describe the type of tools that support this Best Practice. Actual tools may be listed as the Best Practice matures.

Good quality assurance practices will utilize tools to facilitate the production of quality software.

6.1 Detailed Design

An effective design tool will aid in the transition from requirements to design, and can be used to tie together use case creation, code generation, and test case creation during the Development Process. It will also allow the migration of changes to requirements into the design. A design tool that allows integration across Requirements Definition and the Development Process means that all participants will be working from a more consistent framework, and will also allow for easier validation of delivered software.

6.2 Implementation

6.2.1 Source Management

An effective source management system will allow the vendor to track changes to all source code modules that make up their product. It will also enable the vendor to reproduce the software at various milestones, and allow for patches to be created against those releases. Source management tools can allow for parallel development in cases where multiple deliverables need to coexist in a single code base without interfering with each other.

Additionally, source management tools can be used to track the build tools, third-party libraries, and operating systems that make up the environment in which their product is manufactured.

6.3 Internal Testing

6.3.1 Automated Testing and Test Management

Whether a third-party or an internally developed solution, any testing tool will facilitate execution and analysis of test cases, and hence the validation of software. Automated testing allows for easier regression testing, enhancing software quality. Test management allows for reporting and analysis in a centralized manner.

6.4 Configuration/Release Management

An effective release management system will allow for the accounting of all shipments of a product during the Development Process, and track their state through delivery to production installation.

6.5 Problem Reporting

An effective problem reporting and tracking system will enable tracking of all perceived system defects and issues, allowing all affected participants to work together throughout the product development lifecycle. It typically will allow access and entry remotely (enabling customers to view progress of their defects), and track defects and issues from inception through installation. The system can represent a codification of the entire product development lifecycle, and be used to enforce, or assist in enforcing, best practices throughout this lifecycle.

6.6 Tool Integration

The functionality of any of these tools is enhanced by the ability to exchange data with other tools. Many software vendors offer integrated packages that tie together many of the tools used in Requirements Definition and the Development Process.

7 Conformance Overview

Defining conformance and creating a certification policy and program for this Best Practice is the next step in establishing an effective Best Practice. Without the associated conformance criteria and certification processes, there is no assurance that a practitioner has implemented practices according to the approved Best Practice.

Certification provides formal recognition of conformance to an industry Best Practice or Technical Standard specification, which allows:

- Suppliers and practitioners to make and substantiate clear claims of conformance to a Technical Standard or Best Practice
- Buyers to specify and successfully procure from vendors who conform to the Best Practice or provide products that conform to the Technical Standard

Following the approval of this Best Practice, the NSI will work with The Open Group to establish conformance criteria and define an associated Certification Program for this Best Practice. The conformance assessment to be used will be determined at that time. Conformity assessment is the act of determining the compliance of an implementation to a specification, or the adherence of a business operation to a best practice or process definition. There are many techniques for assessing such compliance, including the use of a standardized test method, quality assessment by industry experts, and vendors' claims of conformance made within a defined legal framework. The techniques to be used will be chosen during the process of defining the Certification Program.

Following implementation of the Certification Program, practitioners wishing to have their practices certified as compliant to the Best Practice will be to able to apply for certification of their practices, where a conformance assessment will be performed.

APPENDIXES

A Requirements Checklist

	Requirement	Level	Practitioner	Reference
Development Process: Detailed Design				
1	The vendor must have a documented process for performing detailed design	Must	Vendor	4.3.1.1
2	The documented process should include all of the following: • Standards for the format and content of the design description • Traceability to the system design description and requirements • Technical review of the detailed design • Procedures for tracking and managing changes to the design throughout the product development lifecycle (see Section 4.3.6 (Change Control Management) for further information) • Procedures for ensuring that design constraints or other areas of risk have been considered and assessed	Should	Vendor	4.3.1.1
3	Detailed design must be performed in accordance with the vendor's documented design process.	Must	Vendor	4.3.1.1
Development Process: Implementation				
4	The vendor's approach to implementation must be documented.	Must	Vendor	4.3.2.1
5	Any processes or technical criteria used to support the implementation process, such as checklists, should be formally documented.	Should	Vendor	4.3.2.1

	Requirement	Level	Practitioner	Reference
6	For software component implementation, the software development practices employed should include one or more of the following practices: • Coding is done in accordance with the vendor's documented coding standards. • Code walkthroughs, inspections, or reviews are performed. • Formal methods are used as a software development method to establish the correctness of the implementation of each requirement. • Other specified methods are used to ensure the correctness of the implementation.	Should	Vendor	4.3.2.1
Development Process: Internal Testing				
7	A vendor's internal testing must be performed according to a defined testing process.	Must	Vendor	4.3.3.1
8	The defined testing process should include: • Test planning • Test specification • Test execution • Test reporting	Should	Vendor	4.3.3.1
9	The defined testing process may include: • Root cause analysis	May	Vendor	4.3.3.1
10	The internal testing process should be based on, and closely tied to, the specifications for the system and its various components.	Should	Vendor	4.3.3.1
11	The internal testing process must be documented in a vendor's internal test plan.	Must	Vendor	4.3.3.1
12	Internal testing must be executed in accordance with this test plan.	Must	Vendor	4.3.3.1
13	The vendor's internal test plan must define the method and processes that the vendor will employ during internal testing to assure that the requirements have been correctly implemented and that the system is suitable for delivery to the lottery.	Must	Vendor	4.3.3.1

	Requirement	Level	Practitioner	Reference
14	The vendor's internal test plan must cover: • How the vendor intends to test each new component, and each significant software or hardware change • All levels of testing to be performed as appropriate for the scope of the deliverables; this should include, but is not limited to: – Unit testing – Integration testing – System testing – Regression testing • For each level of testing, the specific types of testing to be performed; Appendix C defines the various types of testing options that may be employed during system testing • Exit criteria that must be fulfilled in order for the vendor to declare the system components suitable for delivery to the lottery • Test reporting that summarizes the results of testing must include: – Any deviation between observed and expected behavior of the system components – Any defects identified	Must	Vendor	4.3.3.1
15	Final system testing should not be performed by the same people who designed the hardware or wrote the software being tested.	Should Not	Vendor	4.3.3.1
Development Process: Acceptance Test Readiness Review				
16	Representatives from the vendor and lottery organizations must hold a review meeting to determine whether the product is ready to transition to Acceptance Testing.	Must	Lottery Vendor	4.3.4.1

	Requirement	Level	Practitioner	Reference
17	The acceptance test readiness review meeting must include the following: • Vendor's test summary report, which must include: – A summary of the types of testing performed – A summary of the results of that testing • Open defects and issues • Lottery's Acceptance Test Plan, which must cover: – Whether the lottery's entry criteria for Acceptance Testing have been fulfilled – Training readiness of the lottery with respect to installation and operation of the system – What the responsibilities of each party are during Acceptance Testing • A definition of the acceptance test environment, which must define exactly what is required to conduct Acceptance Testing	Must	Lottery Vendor	4.3.4.1
18	At the conclusion of the readiness review, either the lottery must authorize the handover for Acceptance Testing, or the vendor and lottery must agree what steps are required for the product to be approved for Acceptance Testing.	Must	Lottery Vendor	4.3.4.1
Development Process: Release Process				
19	The plans for the release of the system or system components must be documented.	Must	Vendor	4.3.5.1
20	The documentation should address the following: • Means of delivery • Delivery medium • Lottery acceptance process and acceptance test period • Knowledge transfer from the vendor to the lottery regarding the use and operation of the system • Problem reporting mechanism to be used after delivery; that is, during Acceptance Testing and post-installation (integration). In some lottery domains, the problem report is referred to as a Testing Incident Report (TIR). • Release notes	Should	Vendor	4.3.5.1

	Requirement	Level	Practitioner	Reference
21	The release notes should include at a minimum: • A formal build description for the software portion of the release, which must include: – The system version – The version of all component modules – The version of all deliverable documents • A list of any known problems in the release and the plans for resolving them • Any limitations in the scope of the system, or any requirements that were included in the specifications but were deferred or eliminated and will not be implemented as a part of this release • A definition of any additional functionality that is being delivered but was not defined in the Requirements Specification (i.e. enhancements); this must include a reference to where the functionality is documented • A list of all deliverables, which must include: – A description of and identifying serial and part numbers for all hardware components – Software components and associated software modules – Documents – Tools • Test report that summarizes the test results of internal testing • Any known usability constraints (for example, performance limitations) • Any known portability constraints • Any known interoperability constraints	Should	Vendor	4.3.5.1
22	The lottery and the vendor must work together to produce an integration plan.	Must	Lottery Vendor	4.3.5.1
23	The lottery and vendor should produce the integration plan during Requirements Definition.	Should	Lottery Vendor	4.3.5.1

	Requirement	Level	Practitioner	Reference
24	The integration plan should account for: • Reviewing the business processes impacted by the changes • Ensuring that people responsible for the affected business functions know of the changes • Ensuring that the people responsible for the affected business functions are notified in advance of what business process changes need to be made (if any) • Ensuring that these non-software issues are resolved in time to support the installation of the new software or hardware	Should	Lottery Vendor	4.3.5.1
25	The integration plan must include information that covers the complete environment into which the product will be integrated and deployed.	Must	Lottery Vendor	4.3.5.1
26	This information must identify the impact of the product release on the overall lottery system, including but not limited to the following: • If any security changes are required, the security changes must be identified in the integration plan together with assigned actions for the responsible parties from either the vendor or the lottery. • If any data conversion is necessary the vendor must provide details and assigned actions for execution of the data conversion. • Identification of platform variations that may occur at different lottery or retail sites and document assigned actions to address them. • Identification of deployment variations that may occur at different lottery or retail and document assigned actions to address them	Must	Lottery Vendor	4.3.5.1
27	The integration plan must be approved by both the lottery and vendor.	Must	Lottery Vendor	4.3.5.1
28	The integration plan may be refined during Acceptance Testing.	May	Lottery	4.3.5.1
29	Integration must be performed in accordance with the agreed plan.	Must	Lottery Vendor	4.3.5.1
30	For integration purposes the vendor must: • Provide an installation guide that is unambiguous and includes installation instructions as well as instructions for backup and back-out procedures where appropriate. • Thoroughly test the installation procedures defined in the installation guide prior to release.	Must	Vendor	4.3.5.1

	Requirement	Level	Practitioner	Reference
Development Process: Change Control Management				
31	The vendor must have a documented change control management process that defines the requirements and process for making modifications to items under change control.	Must	Vendor	4.3.6.1
32	The vendor's documented change control management process should address all the following aspects of change and configuration control: • Identification of the items under change control • The relationship between problem tracking and change control • Mechanisms for change control (e.g., most software development projects use an automated system such as CVS) • Document modification (in particular, it is essential that modifications to requirements are tracked) • Build status tracking (i.e., the relationship between module version and the system version) • Contract modification (a special case of change control requiring formal approval by both the lottery and vendor) • Criteria for approval of different change control elements	Should	Vendor	4.3.6.1
33	The change control management process should include procedures for submitting change requests and tracking the progress of a change request.	Should	Vendor	4.3.6.1
34	A change request must include the following: • A unique tracking number provided by the change management system • Description of the change requested • Reason for the request • Date of the request • Who requested the change • Component or system which is effected	Must	Vendor	4.3.6.1
35	The change request should pass through defined stages as it is being addressed.	Should	Vendor	4.3.6.1

	Requirement	Level	Practitioner	Reference
36	The change request tracking mechanism must provide all relevant parties with access to the status of defects which: • In the case of products under development, are adversely affecting milestone delivery dates or: • In the case of products in the field, are adversely affecting operations	Must	Vendor	4.3.6.1
37	When a change request is approved that requires software modification, the change control process should require use of a software and documentation modification tracking mechanism.	Should	Vendor	4.3.6.1
38	The change control tracking mechanism must require certain information including: • Module • Version • Code modification linked to the version	Must	Vendor	4.3.6.1
39	The change control tacking mechanism may include additional information including: • Check out date • Engineer • Size • Check in date	May	Vendor	4.3.6.1
40	The change control process should require that code modifications be versioned and documented before they are checked back into a library.	Should	Vendor	4.3.6.1
41	If a source code modification alters the system functionality, the change control process should require that the system functional specifications be updated and versioned.	Should	Vendor	4.3.6.1
Development Process: Problem Reporting				
42	The problem reporting process must provide a mechanism that allows each issue to be submitted in a standardized format with standard content, and provides all relevant parties with access to the most current data.	Must	Vendor	4.3.7.1
43	Each issue must have a unique tracking identifier.	Must	Vendor	4.3.7.1

	Requirement	Level	Practitioner	Reference
44	Submission of an issue must include the following information: • Description of the issue • Component or system in which the issue was found. • Severity of the problem • Date the issue was submitted • The submitter of the issue • Phase in which the issue was reported, such as vendor internal testing, acceptance testing, or production • Product version in which the issue was found	Must	Vendor	4.3.7.1
45	Information included in the submission of an issue may include: • Suggestion of how to recreate the issue; for example, pay a ticket after the pay out period expiration	May	Vendor	4.3.7.1
46	The problem reporting process must include a defined process for reviewing and resolving problems.	Must	Vendor	4.3.7.1
47	A team of representatives from each key party should perform review of problem reports.	Should	Vendor	4.3.7.1
48	This team may include representatives from: • The lottery quality assurance team • The development team • The project management	May	Vendor	4.3.7.1
49	Each problem submission should be reviewed by the review team to determine whether it requires immediate action or if it may be deferred to a future release.	Should	Vendor	4.3.7.1
50	The review should consider all of the following: • The overall risk associated with fixing the problem (this may increase or decrease risk) • The cost associated with fixing the problem • The impact on the lottery system or system component associated with fixing the problem	Should	Vendor	4.3.7.1
51	The problem reporting process should include mechanisms that prevent any one individual or group from moving an issue from initial submission to closure.	Should	Vendor	4.3.7.1
52	To facilitate this, a problem report should flow through distinct, defined stages.	Should	Vendor	4.3.7.1
53	At the completion of each stage, the responsible individual should update the problem report with the action taken and a time and date stamp. This ensures that the current status of any issue is available at any time.	Should	Vendor	4.3.7.1

	Requirement	Level	Practitioner	Reference
54	The problem reporting mechanism should provide a means to track issues and their resolution to ensure that critical problems found during one stage are fixed and integrated prior to delivery to the next phase.	Should	Vendor	4.3.7.1
55	The problem reporting process must ensure that the fixes are integrated following the change management system guidelines.	Must	Vendor	4.3.7.1

B Documentation Checklist

This Appendix summarizes the various documentation responsibilities of each party.

Under Responsibility, the following terms are used with these associated meanings:

Sole — For documents in which the specified party has sole responsibility for producing the document in accordance with the requirements of this Best Practice.

Primary — For documents that are to be authored by both parties, this identifies the party with the lead authoring role, and who has overall responsibility for producing the document in accordance with the requirements of this Best Practice.

Secondary — For documents that are to be authored by both parties, this identifies the party that will work with the lead author to produce the document. The Secondary role has the responsibility to provide inputs, author portions of the document, and collaborate with the lead author to ensure successful completion of the document.

Lottery Requirements

Item to be documented	Responsibility	Comments
Integration Plan	Secondary	
Definition of acceptance test environment	Primary	For review with vendor during Acceptance Test Readiness Review. Scope of role will depend on the lottery's specific environment and their relationship with their vendor.

Vendor Requirements

In addition to the specific project documentation identified below, the vendor must have documented product development processes and procedures. The process documentation must cover the following:

- Detailed design process
- Implementation approach
- Internal testing process
- Change control management process
- Problem reporting process

Item to be documented	Responsibility	Comments
Internal Test Plan	Sole	
Test Summary Report	Sole	For review with lottery during Acceptance Test Readiness Review.
List of open defects and issues	Sole	For review with lottery during Acceptance Test Readiness Review.
Definition of acceptance test environment	Secondary	For review with lottery during Acceptance Test Readiness Review. Scope of vendor's role will depend on specific customer's environment and needs.
Release Plan & Release Notes	Sole	
Integration Plan	Primary	
Installation Guide	Sole	

C System Testing

This appendix defines the various types of testing that may be deployed during the testing of lottery systems. System testing is usually performed during a vendor's internal test process and during the lottery's Acceptance Testing.

The test methods defined below are applicable for use in testing a complete lottery system. The descriptions provide information on what the test method is, the purpose of that particular type of testing, and techniques for how the testing is performed.

C.1 Anomaly Testing

Anomaly testing is used to determine how the system reacts to anticipated user errors such as invalid input. This testing will help validate that error messages are useful and accurate.

C.2 Business Cycle Testing

Business cycle testing is used to verify the operation of the system over time. This testing emulates the activities performed on the system over all applicable business cycles, including daily, weekly, and monthly cycles, and any events that are date-sensitive.

This testing is performed by identifying a time period, such as an invoice period, and executing all transactions and activities that would occur during that period.

C.3 Configuration Testing

Configuration testing verifies the operation of the system on multiple platform configurations. This type of testing is intended to uncover compatibility issues between different software and hardware configurations. In most production environments, the particular hardware specifications for the client workstations, network connections, and database servers vary. Client workstations may have different software loaded; for example, applications or drivers, and at any one time, many different combinations may be active using different resources.

C.4 Conversion Testing

Conversion testing is used to verify that data is handled consistently when converting from one system to another (i.e., converting to a new gaming system.). This is accomplished by running data through both systems in parallel and validating that the systems show the same results.

C.5 Failover and Recovery Testing

Failover and recovery testing verify that the system can successfully failover and recover from a variety of hardware, software, or network malfunctions with undue loss of data or data integrity.

Failover testing ensures that, for those systems that need to be kept running, when a failover condition occurs, the alternate or backup systems properly "take over" for the failed system without loss of data or transactions.

Recovery testing is an antagonistic test process in which the system is exposed to extreme conditions, or simulated conditions, to cause a failure, such as device input/output (I/O) failures or invalid database pointers and keys. Recovery processes are invoked and the system is monitored and inspected to verify proper system and data recovery has been achieved. Recovery testing needs to cover both the automated aspects of system recovery as well as the manual procedures required.

C.6 Functional Testing

Functional testing is testing of a system against its base requirements. This type of testing is based upon black box techniques in which the tester knows the inputs and expected outcomes of the system, but not how the program arrives at those outputs.

The purpose of functional testing is to verify that the system performs in accordance with the specified business and technical requirements. The goal of functional testing is to verify system functions such as proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules.

Functional testing verifies the system or component and its internal processes by interacting with the system via the user interface and analyzing the output or results. Functional testing is used to verify that the system performs correctly when subjected to a variety of circumstances and repetition of the transactions.

One of the specific aspects of functional testing important to the testing of lottery systems is included below.

Audit Trail Testing

Audit trail testing is testing of the audit trail function to ensure that a source transaction can be traced to a control total, that the transaction supporting a control total can be identified, and that the processing of a single transaction or the entire system can be reconstructed using audit trail information.

C.7 Installation Testing

There are two types of installation testing.

The first is typically used by vendors when preparing a system for release to a customer. The purpose of this testing is to ensure that the system can be installed under different conditions

such as a new installation, an upgrade, and a complete or custom installation under normal and abnormal conditions. Abnormal conditions include insufficient disk space, lack of privilege to create directories, and so on.

The second type of installation testing, which may be performed during either the Development Process or Acceptance Testing, verifies that, once installed, the system operates correctly. This usually means running a number of the tests that were developed for functional testing.

C.8 Interoperability Testing

Interoperability testing is a formalized testing process where people, procedures, and systems/equipment are brought together in an operational environment to test the system interfaces and determine the reliability, usability, timeliness, and accuracy of the exchanged information.

In the lottery environment, interoperability testing is primarily testing that a lottery system interacts with other systems, such as a credit card authorization system, an ICS, or another back-office system. Testing is performed at the system boundaries to make sure that the two systems interface correctly.

C.9 Operations Testing

Operations testing is the testing of a complete system's operational characteristics and processes including start-up, operation, and recovery.

Operations testing verifies that the system can be operated and supported by the operations staff in an efficient and consistent manner. Testing is usually performed following documented operational procedures and checklists. Operations testing is performed on the production system, or a system that mimics the production system.

C.10 Performance Testing

Performance testing is designed to establish the performance of a system against predefined metrics or other alternative systems. Performance testing will test aspects of a system's performance such as response times, transaction rates, availability, capacity, and scalability. Typical performance testing measures include throughput, response time, storage capacity, and concurrent use. There are multiple types of performance-related tests; each is described below.

Performance Profiling

Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated. The goal of performance profiling is to verify that performance requirements have been achieved. Performance profiling is implemented and executed to profile and tune a system's performance behaviors as a function of conditions such as workload or hardware configurations.

Load Testing

Load testing is a performance test which subjects the system to varying workloads to measure and evaluate the performance behaviors and ability of the system to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics such as response times, transaction rates, and other time-sensitive issues.

Stress Testing

Stress testing is a type of performance test implemented and executed to find errors due to low resources or competition for resources. Low memory or disk space may reveal defects in the system which are not apparent under normal conditions. Other defects might result from competition for shared resources like database locks or network bandwidth. Stress testing can also be used to identify the peak workload the system can handle.

Stress testing is very important given the potential for extreme spikes in system use during high jackpot periods and other events that place unexpected loads on the system. Stress testing should occur at various times throughout the internal test cycles and, if required, during the acceptance test cycle. Stress testing should emulate various scenarios and operating conditions to ensure the system will degrade gracefully. In addition to exercising normal usage patterns (i.e., wagers, validations, cancellations), stress testing should also exercise any system mechanisms to provide point-of-sale updates as these can place extreme load on the communication mechanism.

C.11 Regression Testing

Regression testing is the selective re-testing of a system or component that has been modified to verify that the modifications have not caused unintended effects and that the system or component still complies with its specified requirements.

In the context of a particular system component, regression testing is used to ensure that modifications to the system component to fix defects or add functionality have not introduced problems in unmodified and previously working functions of the component.

In the context of a lottery system, regression testing is used to ensure that the system component being installed does not affect any portion of the lottery system already installed or any system components that interface with the new component.

Regression testing is typically performed using previous error-free tests to provide the assurance that the defects reported to be fixed are indeed fixed and the system or component has not introduced unintended effects in the unaltered parts of the system. It is considered best practice to automate regression testing wherever possible.

C.12 Security and Access Control Testing

Security and access control testing is performed to ensure that established security rules, procedures, or regulations are properly handled by the system.

Security and access control testing focus on two key areas of security:

- Application-level security, including access to the data or business functions
- System-level security, including logging into or remote access to the system

Application-level security ensures that, based upon the desired security, actors are restricted to specific functions or use cases, or are limited in the data that is available to them. For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that "user one" can see all customer information, including financial data; however, "user two" only sees the demographic data for the same client.

System-level security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.

Virus protection and intrusion detection ensure that systems are not susceptible to unwanted access and control. The system should be tested to ensure that intrusions and viruses are not facilitated; for example, by leaving open ports.

C.13 Usability Testing

The ultimate success of the system will depend heavily on the ability of people to use it. Testing the ease-of-use of the system by people is an important aspect of testing. As usability is difficult to evaluate prior to the test phases, it is important that the people aspect of the system is evaluated in as realistic an environment as possible. One aspect of usability testing is manual testing of typical usage scenarios to ensure that the people interacting with the automated system can perform their functions correctly.

Testing for "user- friendliness" is clearly subjective and will depend on the targeted end user or customer. User interviews, surveys, video recordings of user sessions, and other techniques may be used. Quality assurance testers are usually not appropriate usability testers. Quality assurance typically measures the degree to which the system can be understood, learned, used, and liked by the user when the system is used under specified conditions. Quality assurance tests the ease-of-navigation, layout and design, performance, error feedback, and consistency of the system to determine the system 's overall usability.

If a User Guide accompanies the system, it is reviewed to verify that all instructions are correct and that all figures and images displayed in the User Guide match the screens displayed in the system. As many users rely on the User Guide that accompanies a system, it is critical to ensure that it is correct.

C.14 Volume Testing

Volume testing subjects the system to large amounts of data to determine whether limits are reached that cause the software to fail. Volume testing also identifies the continuous maximum load or volume the system can handle for a given period.

For example, if the system is processing a set of database records to generate a report, a volume test would use a large test database and check that the software behaved normally and produced the correct report.

Glossary

The following terms and acronyms are used in this document.

Lottery Industry-Specific Terminology

Term or Acronym	Definition
Administrative Systems	Systems designed to support the lottery's business operations, such as accounting, retailer management, claims processing, and information management.
Back Office Systems	Data processing systems used to support the central business operation of the lottery, as distinct from gaming systems or systems employed by the retailer at the point of sale.
Gaming System	The set of software and hardware components required to deploy a particular game or set of games, which includes game terminals, network, and game host computers at the lottery central office.
ICS Vendor	The entity producing the Internal Control System, which may be a third-party vendor or the Lottery Development Organization.
Internal Control System (ICS)	The audit system and its associated processes, which perform auditing of the Gaming System component to ensure the integrity, security, and accuracy of gaming transactions.
Lottery Development Organization	The group within the lottery that is responsible for the development and/or integration of software and hardware components that comprise the lottery systems. They are responsible for running the IT systems and to a certain extent they act as an in-house technical development team. This service is often outsourced to third-party vendors.
Lottery Environment	The full set of software and hardware components that comprise a lottery, including gaming systems, the ICS system, administrative and back office systems, website, instant tickets, telecommunications network infrastructure, as well as the human participants who operate the hardware and software components including the Lottery Organization, vendors, retailers, and players.
Lottery Organization (Lottery)	The lottery organization comprises all those responsible for the overall operation of the lottery, which includes the director and other management personnel, and operational and technical personnel including the lottery development office. Together, they are responsible for overseeing lottery integrity, optimizing profitability from games (current and future), system procurement, managing retailers and vendors, and for marketing and payouts.

Term or Acronym	Definition
Lottery System (System)	The software and hardware associated with a particular function within the lottery, such as a gaming system.
Product	Product refers to the software and/or hardware that the lottery has contracted the vendor to produce.
System Components	The software and hardware components associated with a lottery system.
System Vendor	The entity producing the system components, which may be a third-party vendor or the Lottery Development Organization.

General Software and Hardware Industry Terminology

Term or Acronym	Definition
Acceptance Testing	Testing performed by the lottery to determine the acceptability of the delivered lottery system or system components for deployment into the lottery environment.
Code Module	A code module consists of lines of high-level computer language which, when compiled, are designed to carry out a specific function or sub-function of the system. A code module will include self-contained documentation for maintainability in the form of explanatory comment text placed within the lines of code themselves and at the beginning of the module.
Code Walkthrough	A technical review, normally by peers, in which a programmer's code is reviewed by other programming professionals with the intent to identify bugs at an early point and prior to system integration.
Formal Method	The use of a mathematical model or fourth generation language for software design such that a proof of correctness of the resultant code to the requirement from which it was generated is inherent, thus dispensing with the need for internal testing to assure quality.
Integration Testing	Testing in which software components, hardware components, or both, are combined and tested to evaluate the interactions between them. Integration testing focuses on testing the interfaces between the components.
Interoperability	The exchange of data between separate heterogeneous systems.
Portability	The ability of software application code to be run on heterogeneous platforms without change to the code.
Requirements Specification	The Requirements Specification documents a description of the expected features, constraints, interfaces, and other attributes of the system components to be produced, and forms the basis for the agreement between the lottery and the system vendor on what the

Term or Acronym	Definition
	system is designed to do.
Sign-off	The act of formal agreement that a given phase in the product development lifecycle is complete.
System Design Specification	The System Design Specification, when used, is the document in which the system design description is documented. The system design description provides a high-level description of the overall lottery system, the software and/or hardware component(s) that comprise it, and how the components interact.
System Testing	Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
Test Case	A Test Case is a documented single test instance that includes a set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific single requirement. It is desirable for test cases to be uniquely identifiable and traceable. Multiple test cases may exist for a single requirement.
Test Scenario	A set of test cases used to validate the behavior of a product through testing of the business process flows.
Test Script	A detailed set of instructions for the set-up, execution, and evaluation of results of one or more test cases. A test script may be executed by an automated test tool or manually.
Test Summary Report	A written report produced at the end of a period of testing (such as vendor internal testing, or the lottery's acceptance testing) that provides a summary of the types of testing performed and a summary of the results of that testing.
Unit Testing	The testing of individual hardware or software units or groups of related units.

Index